

Programação Imperativa – Exame da Época Especial – 4.Set.2015

Parte A

Considere as seguintes definições de tipos:

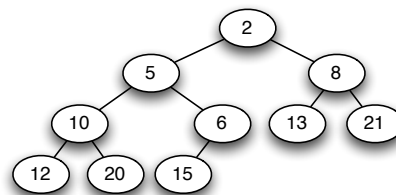
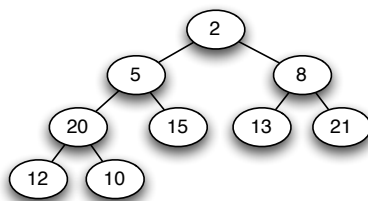
```
typedef struct slist {  
    int valor;  
    struct slist *prox;  
} *LInt;  
  
typedef struct nodo {  
    int valor;  
    struct nodo *esq, *dir;  
} *ABin;
```

1. Defina uma função `int retiraNeg (int v[], int N)` que retira os números negativos de um vector com N inteiros. A função deve retornar o número de elementos que não foram retirados.
2. Apresente uma definição da função pré-definida em C `int strcmp (char s1[], char s2[])` que compara (lexicograficamente) duas strings. O resultado deverá ser
 - 0 se as strings forem iguais
 - <0 se $s1 < s2$
 - >0 se $s1 > s2$
3. Apresente uma definição não recursiva da função `void insert (LInt *, int)` que insere ordenadamente um elemento numa lista ordenada.
4. Defina uma função `void split (LInt l, int x, LInt *mx, LInt *Mx)` que, dada uma lista ligada l e um inteiro x, parte a lista em duas (retornando os endereços dos primeiros elementos da lista em *mx e *Mx): uma com os elementos de l menores do que x e a outra com os restantes. Note que esta função não deverá criar cópias dos elementos da lista.

Parte B

Uma *heap* é uma árvore binária em que o elemento que está na raiz é menor do que todos os outros. Para além disso as subárvores também verificam essa propriedade.

Das árvores que se apresentam abaixo a da direita é uma heap enquanto que a da esquerda não é (note que a subárvore de altura 2 mais à esquerda não é uma heap).



1. Defina uma função `int heapOK (ABin h)` que retorna 1 se a árvore binária é uma heap.
2. Defina uma função `int maior (ABin h)` que, dada uma heap não vazia, retorna o maior elemento.
3. Defina uma função `int removeMin (ABin *h)` que, dada uma heap não vazia, retorna o menor elemento (a que está na raiz) e remove-o da heap. Esta remoção deverá garantir que a árvore resultante é uma heap.
4. Defina uma função `int primeiros (ABin h, int v[], int N)` que dada uma heap h preenche o array v com os elementos da heap por ordem crescente. A função deverá preencher no máximo N posições do array e retornar o número de elementos que foram preenchidos.